ИНСТИТУТ ИНТЕЛЛЕКТУАЛЬНЫХ КИБЕРНЕТИЧЕСКИХ СИСТЕМ

Кафедра
«Криптология и кибербезопасность»

Лабораторная работа №5

по предмету «Технологии контейнеризации»

Выполнил студент группы Б20-505

Сорочан Илья

Москва – 2023

Содержание

Предисловие

Теперь лабораторные выполняются на Windows, потому что Debian 11 часто отбивается от сети университета, из-за чего каждые пять минут приходиться логиниться снова, а после ~5 раз достигается лимит устройств.

1. Подготовка виртуальной машины

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/jammy64"

  config.vm.box_check_update = false

  config.vm.network "public_network"

  config.vm.network "forwarded_port", guest: 53, host: 53
  config.vm.network "forwarded_port", guest: 80, host: 80
  config.vm.network "forwarded_port", guest: 443, host: 443

  config.vm.provider "virtualbox" do |vb|
    vb.memory = "4096"
    vb.cpus = 2
    vb.check_guest_additions = false
    vb.customize ["modifyvm", :id, "--nested-hw-virt", "on"]
  end

  config.vm.provision :docker
  config.vm.provision :docker_compose
end
```

Рис. 1. Конфигурация `vagrant`

В данном `Vagrantfile`:

- отключаются обновления для образа и гостевых дополнений (необходим плагин `vagrant-vbguest`);
- проброс портов, которые могут понадобится в будущем;
- использованеи моста (`public_network`) для доступа к nginx извне;

- общая папка для сохранения файлов Dockerfile, docker-compose.yml и сертификатов;
- ограничения на память (могут понадобиться);
- установка docker и docker-compose (для последнего необходим плагин `vagrant-docker-compose`).

2. Сертификаты

```
vagrant@ubuntu-jammy:~$ sudo apt install mkcert
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  mkcert
0 upgraded, 1 newly installed, 0 to remove and 61 not upgraded.
Need to get 1314 kB of archives.
After this operation, 3342 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 mkcert amd64 1.4.3-1ubuntu0.2 [1314 kB]
Fetched 1314 kB in 1s (1256 kB/s)
Selecting previously unselected package mkcert.
(Reading database ... 70292 files and directories currently installed.)
Preparing to unpack .../mkcert_1.4.3-1ubuntu0.2_amd64.deb ...
Unpacking mkcert (1.4.3-1ubuntu0.2) ...
Setting up mkcert (1.4.3-1ubuntu0.2) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Рис. 2. Установка mkcert

```
vagrant@ubuntu-jammy:~$ mkcert -install
The local CA is now installed in the system trust store! ⚡
```

Рис. 3. Установка корневого сертификата

```
vagrant@ubuntu-jammy:~$ mkcert dns.student registry.student

Created a new certificate valid for the following names
 - "dns.student"
 - "registry.student"

The certificate is at "./dns.student+1.pem" and the key at "./dns.student+1-key.pem"

It will expire on 5 March 2026
```

Рис. 4. Выпуск необходимых сертификатов

```
$mkcert = <<-SCRIPT
sudo apt install mkcert -y
mkcert -install

cd /home/vagrant
mkdir -p ./nginx/certs
chown vagrant ./nginx

mkcert -cert-file ./nginx/certs/dns.student.crt -key-file ./nginx/certs/dns.student.key dns.student
mkcert -cert-file ./nginx/certs/registry.student.crt -key-file ./nginx/certs/registry.student.key registry.student
SCRIPT
```

Рис. 5. Исправление формата и задание как provision

3. Nginx

```
vagrant@ubuntu-jammy:~$ cd /etc/nginx
vagrant@ubuntu-jammy:/etc/nginx$ sudo mkdir certs sites-available sites-enabled
mkdir: cannot create directory ‘sites-available’: File exists
mkdir: cannot create directory ‘sites-enabled’: File exists
vagrant@ubuntu-jammy:/etc/nginx$ mv ~/*.pem certs/.
mv: cannot move '/home/vagrant/dns.student+1-key.pem' to 'certs/./dns.student+1-key.pem': Permission denied
mv: cannot move '/home/vagrant/dns.student+1.pem' to 'certs/./dns.student+1.pem': Permission denied
vagrant@ubuntu-jammy:/etc/nginx$ sudo mv ~/*.pem certs/.
```

Рис. 6. Создание необходимых директорий и перемещение сертификатов

```
  config.vm.provision "file", source: "./app.conf", destination: "~/nginx/conf/app.conf"
  config.vm.provision "file", source: "./docker-compose.yml", destination: "~/docker-compose.yml"
  config.vm.provision "file", source: "Dockerfile", destination: "~/Dockerfile"
```

Рис. 7. Provision

Файл app.conf:

```
server {
    listen 80;
    server_name dns.student registry.student;

    # Перенаправление с HTTP на HTTPS
```

```
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name dns.student;

    ssl_certificate /etc/nginx/ssl/dns.student.crt;
    ssl_certificate_key /etc/nginx/ssl/dns.student.key;

    location / {
        proxy_pass http://dns:3000;  # DNS
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}

server {
    listen 443 ssl;
    server_name registry.student;

    ssl_certificate /etc/nginx/ssl/registry.student.crt;
    ssl_certificate_key /etc/nginx/ssl/registry.student.key;

    location / {
        proxy_pass http://registry:5000;  # Registry
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```

4. Docker compose

```
version: '3'
services:
  dns:
```

```yaml
    image: adguard/adguardhome:latest
    ports:
      - '53:53/tcp'
      - '53:53/udp'
    networks:
      - lab5_network
    restart: unless-stopped
    volumes:
      - volume_dns_work:/opt/adguardhome/work
      - volume_dns_conf:/opt/adguardhome/conf
    deploy:
      resources:
        limits:
          cpus: '0.5'
          memory: 512M
    hostname: dns

  nginx:
    image: nginx:latest
    restart: unless-stopped
    ports:
      - '80:80'
      - '443:443'
    volumes:
      - ./nginx/conf/app.conf:/etc/nginx/conf.d/default.conf:ro
      - ./nginx/certs:/etc/nginx/ssl
    networks:
      - lab5_network
    depends_on:
      - dns
      - registry
    deploy:
      resources:
        limits:
          cpus: '1'
          memory: 1G
```

```yaml
    hostname: web

  registry:
    image: registry:2
    hostname: registry
    networks:
      - lab5_network
    deploy:
      resources:
        limits:
          cpus: '0.5'
          memory: 512M
    environment:
      - REGISTRY_AUTH=htpasswd
      - REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd
      - REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm
    volumes:
      - ./auth:/auth
    restart: unless-stopped
    depends_on:
      - dns

networks:
  lab5_network:

volumes:
  volume_dns_work:
  volume_dns_conf:
  registry_auth:
```

```
ERROR: for vagrant_dns_1  Cannot start service dns: driver failed programming external connectivity on endpoint vagrant_
dns_1 (9a6c9b914dfd0e071a71324df9a3b7efe5a94068734a5bb04de7bb907565f6fb): Error starting userland proxy: listen tcp4 0.0
.0.0:53: bind: address already in use
```

Рис. 8. Первая попытка поднять

```
vagrant@ubuntu-jammy:~$ sudo lsof -i :53
COMMAND   PID            USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
systemd-r 561 systemd-resolve   13u  IPv4  17168      0t0  UDP localhost:domain
systemd-r 561 systemd-resolve   14u  IPv4  17169      0t0  TCP localhost:domain (LISTEN)
```

Рис. 9. Смотрим кто занял 53 порт

```
[Resolve]
# Some examples of DNS servers which may be used for DNS= and FallbackDNS=:
# Cloudflare: 1.1.1.1#cloudflare-dns.com 1.0.0.1#cloudflare-dns.com 2606:4700:4700::1111#cloudflare-dns.com 2606:
# Google:     8.8.8.8#dns.google 8.8.4.4#dns.google 2001:4860:4860::8888#dns.google 2001:4860:4860::8844#dns.goog
# Quad9:      9.9.9.9#dns.quad9.net 149.112.112.112#dns.quad9.net 2620:fe::fe#dns.quad9.net 2620:fe::9#dns.quad9.
DNS=127.0.0.1
#FallbackDNS=
#Domains=
#DNSSEC=no
#DNSOverTLS=no
#MulticastDNS=no
#LLMNR=no
#Cache=no-negative
#CacheFromLocalhost=no
DNSStubListener=no
#DNSStubListenerExtra=
#ReadEtcHosts=yes
#ResolveUnicastSingleLabel=no
```

Рис. 10. Ставим no в /etc/systemd/resolved.conf

```
sudo systemctl restart systemd-resolved
```

```
vagrant@ubuntu-jammy:~$ docker-compose up -d
WARNING: Some services (dns, nginx, registry) use the 'deploy' key, which will be ignored. Compose does not support 'dep
loy' configuration - use `docker stack deploy` to deploy to a swarm.
Starting vagrant_dns_1 ... done
Creating vagrant_registry_1 ... done
Creating vagrant_nginx_1    ... done
```

Рис. 11. Успешный запуск

```
$dns_clear = <<-SCRIPT
sudo sed -i 's/#DNS=/DNS=127.0.0.1/' /etc/systemd/resolved.conf
sudo sed -i 's/#DNSStubListener=yes/DNSStubListener=no/' /etc/systemd/resolved.conf
sudo systemctl restart systemd-resolved
SCRIPT
```

Рис. 12. Provision

5. Настройка DNS

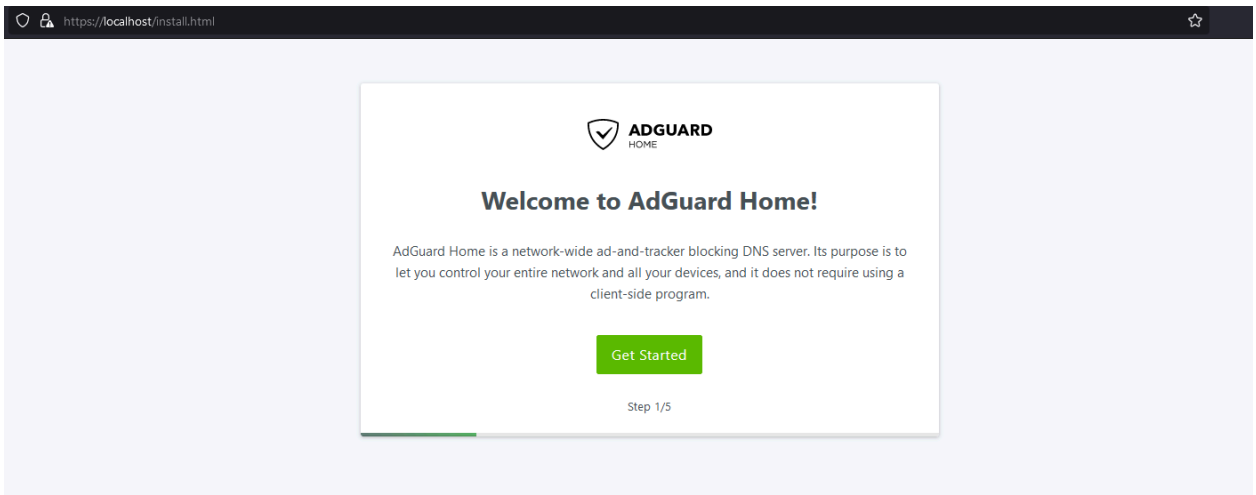


Рис. 13. Приветственное окно

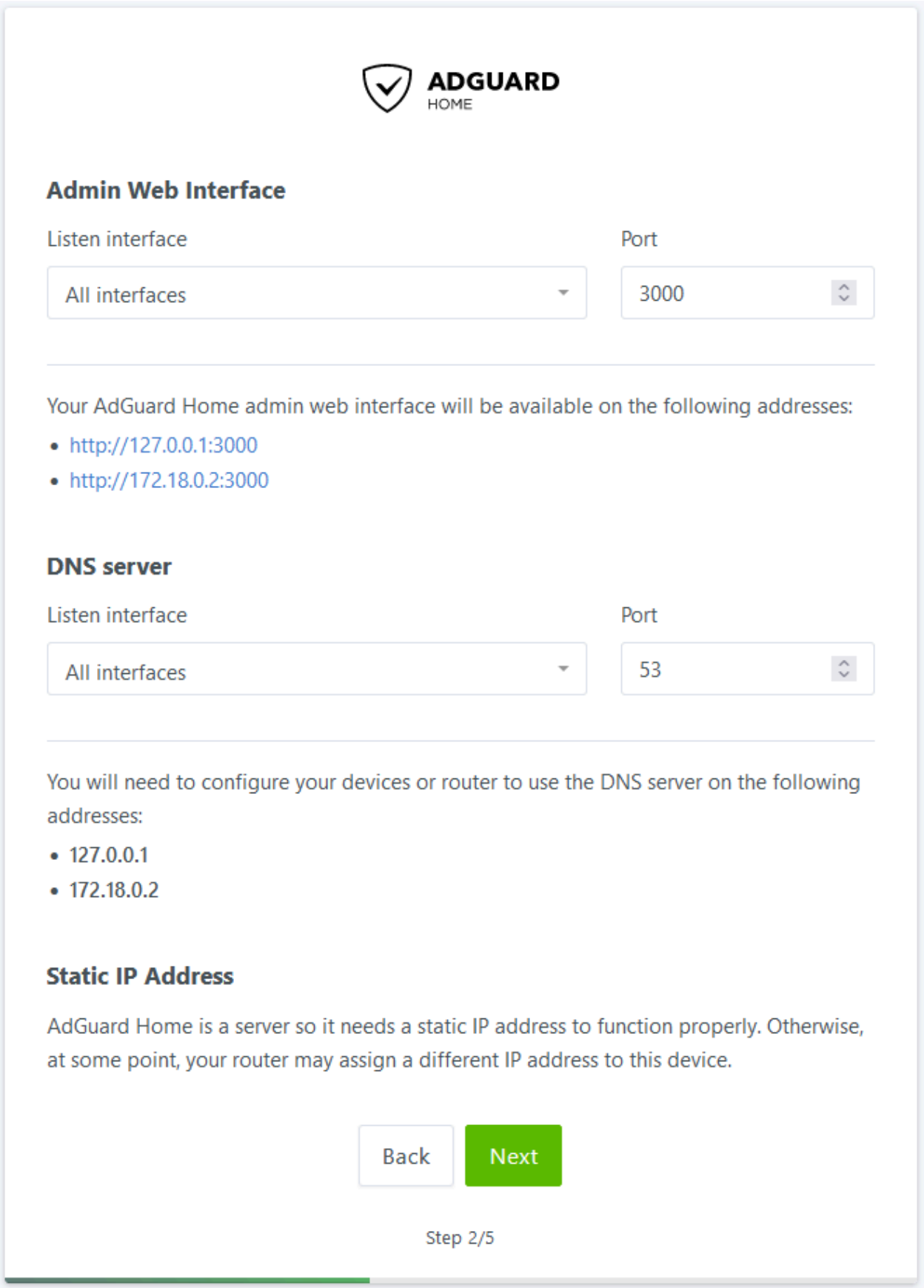


Рис. 14. Ставим порт на 3000

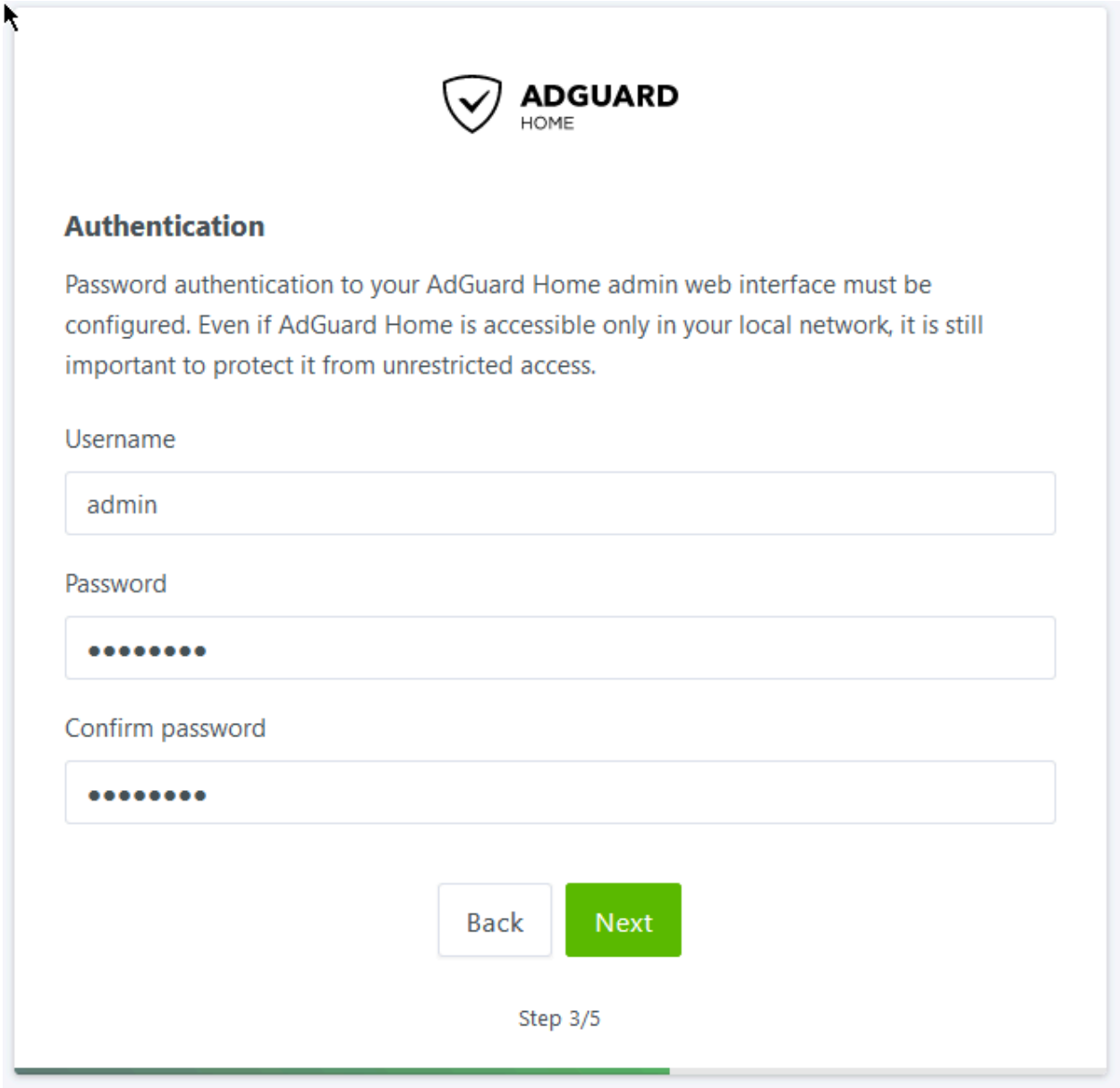


Рис. 15. Устанавливаем пароль

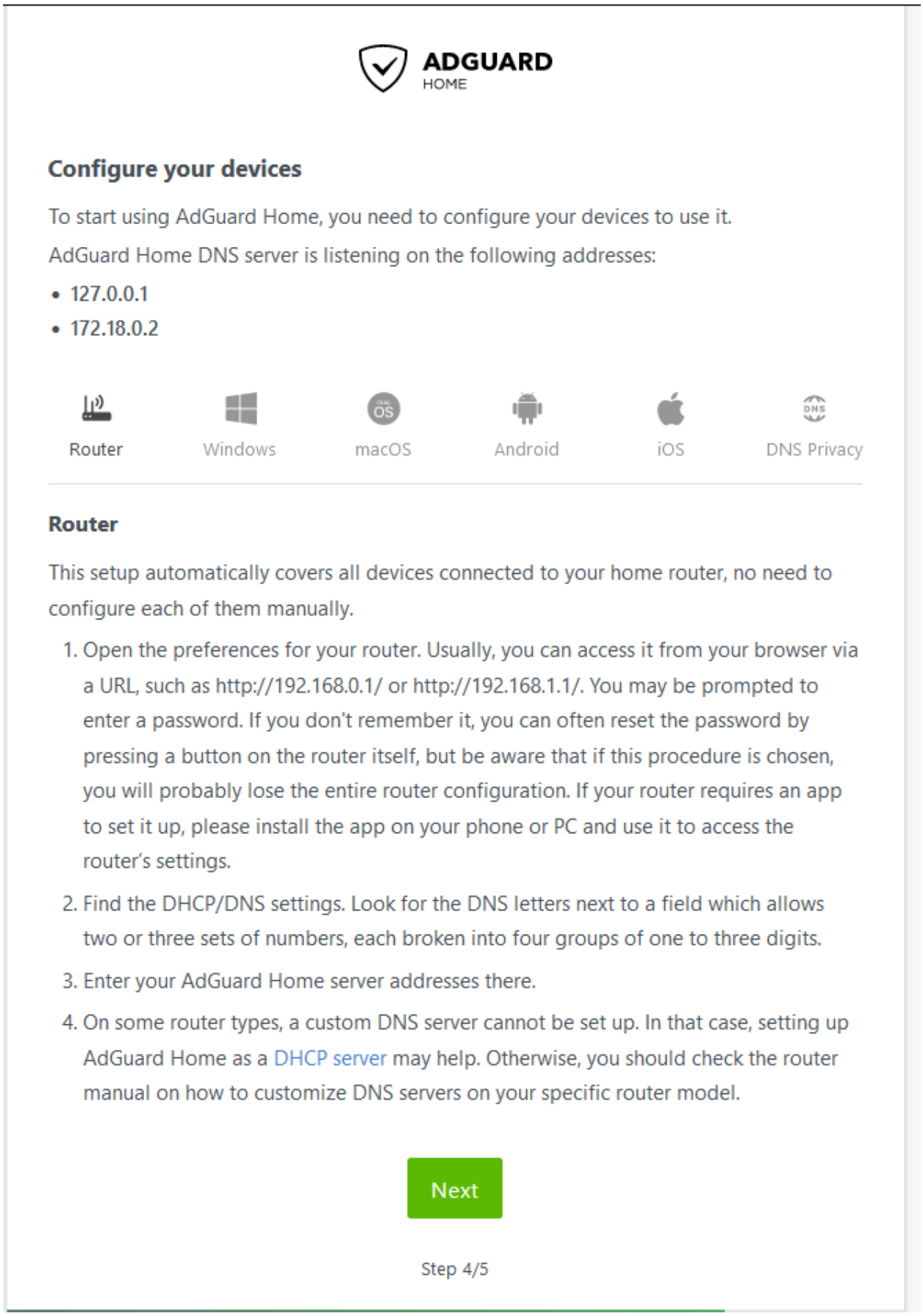


Рис. 16. Жмем next

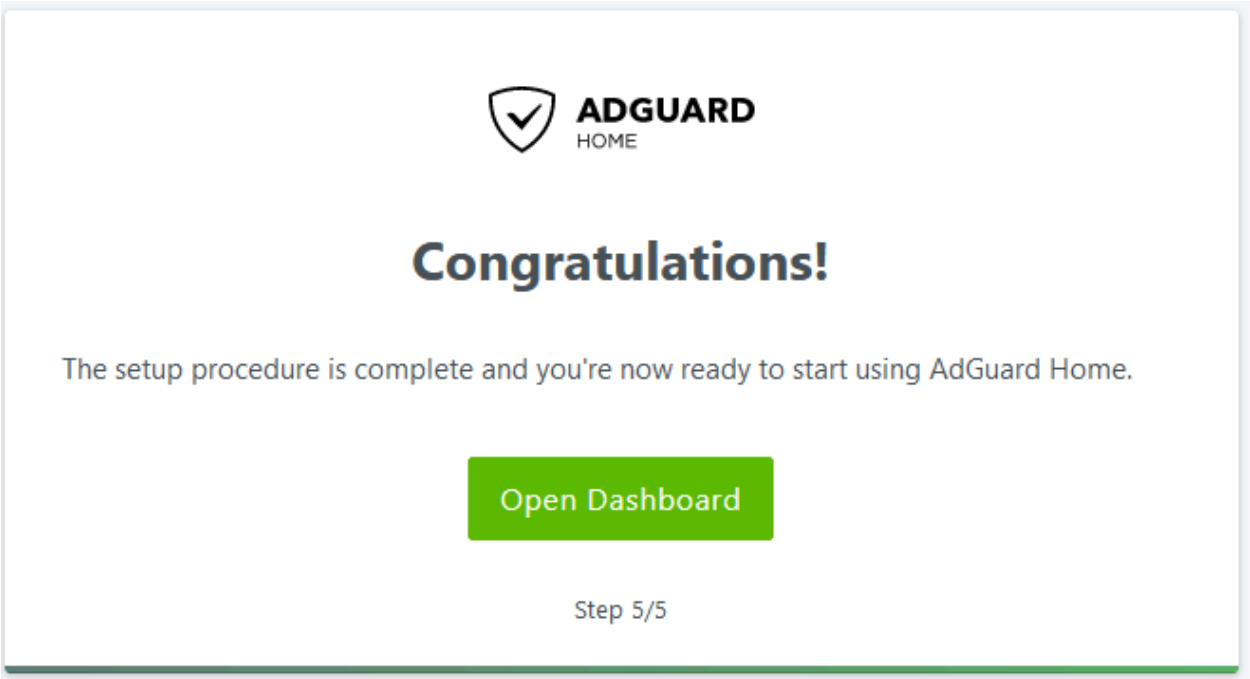


Рис. 17. Завершение

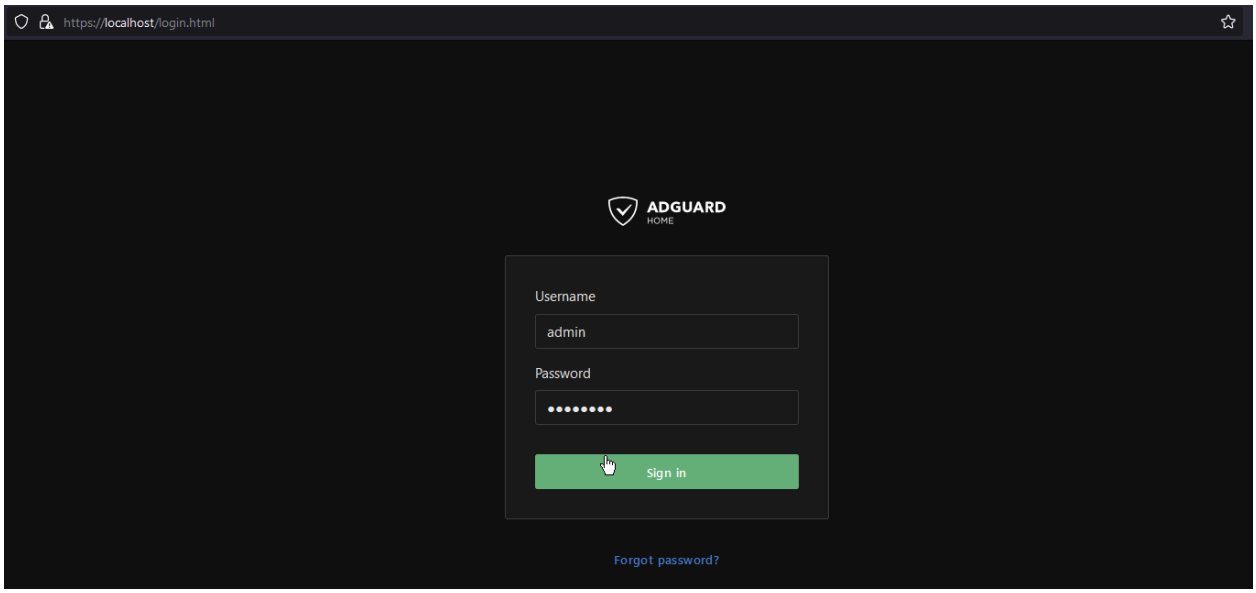


Рис. 18. Открываем dashboard

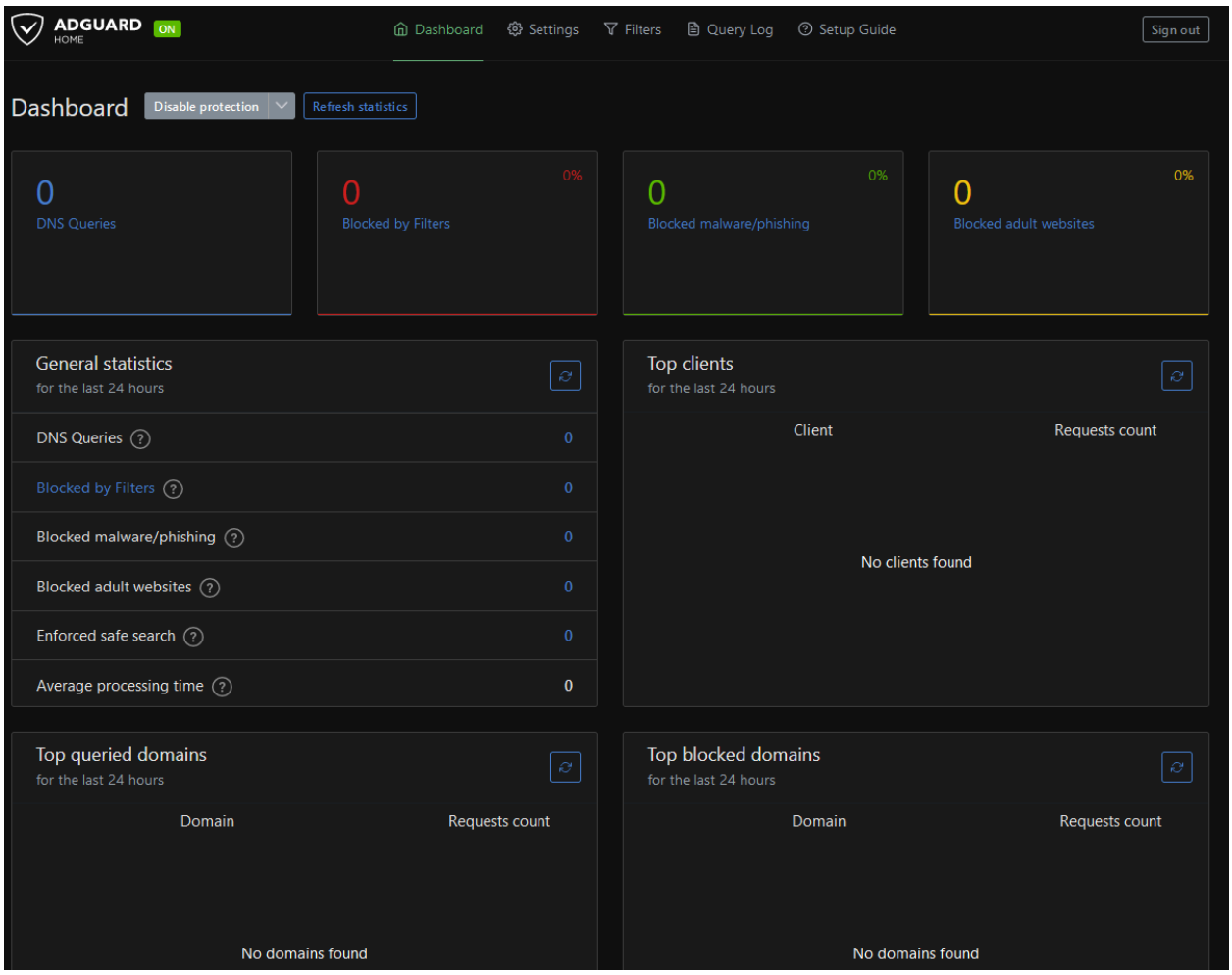


Рис. 19. Dashboard

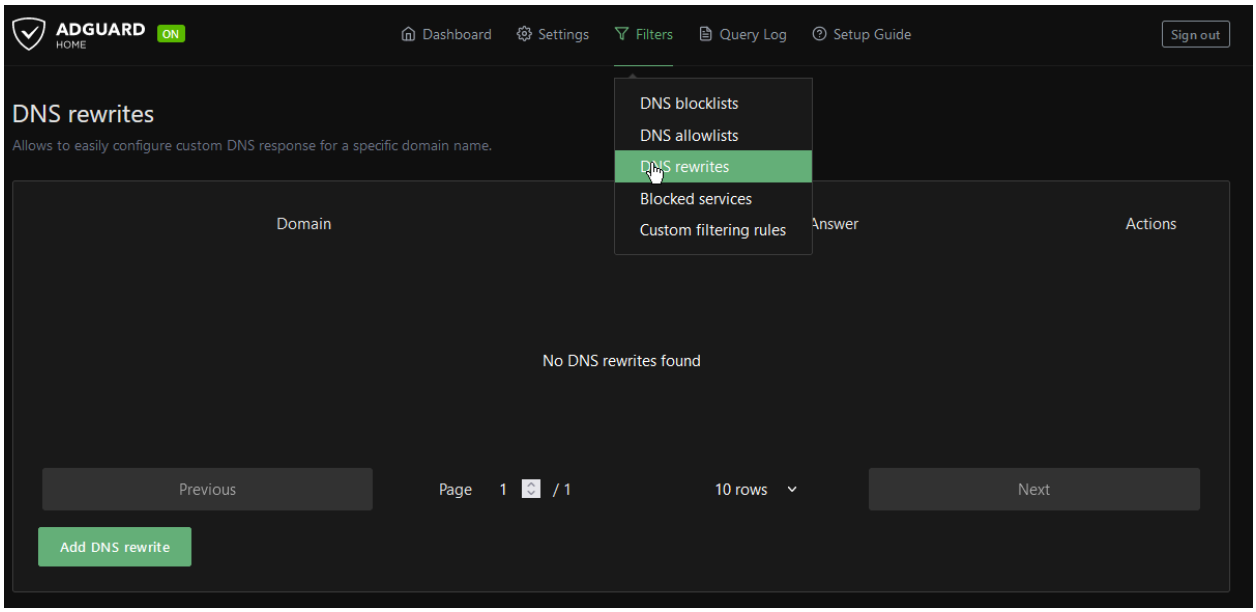


Рис. 20. Перезаписываем DNS (1)

Domain	Answer	Actions
dns.student	127.0.0.1	
registry.student	127.0.0.1	

Рис. 21. Перезаписываем DNS (2)

```
vagrant@ubuntu-jammy:~$ nslookup dns.student
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
Name:   dns.student
Address: 127.0.0.1

vagrant@ubuntu-jammy:~$ nslookup registry.student
Server:         127.0.0.1
Address:        127.0.0.1#53

Non-authoritative answer:
Name:   registry.student
Address: 127.0.0.1
```

Рис. 22. Проверка из виртуальной машины

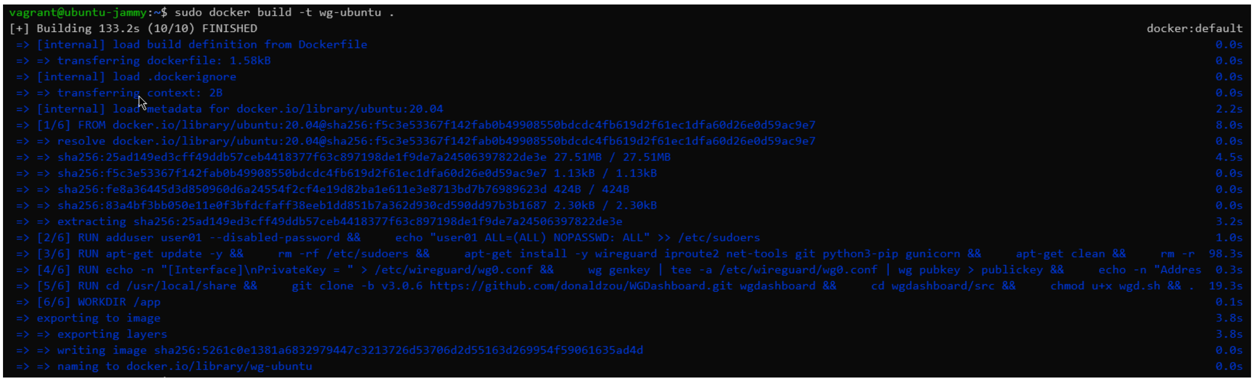
6. Загрузка wg-dashboard

Используюем следующий Dockerfile (из прошлой лабы):

```
FROM ubuntu:20.04

LABEL author="Sorochan I.V."
LABEL email="none"
LABEL version="0.1.0"
LABEL description="wgdashboard + wireguard + ubuntu"

RUN adduser user01 --disabled-password && \
```

```
    echo "user01 ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers

# Update
# Sudoers = hotfix
RUN apt-get update -y && \
    rm -rf /etc/sudoers && \
    apt-get install -y wireguard iproute2 net-tools git python3-pip
gunicorn && \
    apt-get clean && \
    rm -rf /var/lib/apt/lists/*

# Setup wireguard config
RUN echo -n "[Interface]\nPrivateKey = " > /etc/wireguard/wg0.conf
&& \
   wg genkey | tee -a /etc/wireguard/wg0.conf | wg pubkey > publickey
&& \
    echo -n "Address = 10.0.0.1/" >> /etc/wireguard/wg0.conf && \
   ip -of inet addr show eth0 | awk '{split($4, a, "/"); print a[2]}'
>> /etc/wireguard/wg0.conf && \
    echo "ListenPort = 51820" >> /etc/wireguard/wg0.conf && \
    echo -n "\n[Peer]]\nPublicKey = " >> /etc/wireguard/wg0.conf &&
\
    cat publickey >> /etc/wireguard/wg0.conf && rm -f publickey && \
    echo "AllowedIPs = 0.0.0.0/0" >> /etc/wireguard/wg0.conf

# Setup wg-dashboard
RUN cd /usr/local/share && \
    git clone -b v3.0.6 https://github.com/donaldzou/WGDashboard.git
wgdashboard && \
    cd wgdashboard/src && \
    chmod u+x wgd.sh && ./wgd.sh install && \
    sudo chmod -R 755 /etc/wireguard

EXPOSE 51820/udp
EXPOSE 10086/tcp
```

```
WORKDIR /app

ENTRYPOINT ["/bin/bash", "-c", "wg-quick up wg0 && cd /usr/local/
share/wgdashboard/src && ./wgd.sh start && tail -f /dev/null"]
```



```
vagrant@ubuntu-jammy:~$ sudo docker build -t wg-ubuntu .
[+] Building 133.2s (10/10) FINISHED                                                                                                             docker:default
 => [internal] load build definition from Dockerfile                                                                                                       0.0s
 => => transferring dockerfile: 1.58kB                                                                                                                     0.0s
 => [internal] load .dockerignore                                                                                                                          0.0s
 => => transferring context: 2B                                                                                                                            0.0s
 => [internal] load metadata for docker.io/library/ubuntu:20.04                                                                                            2.2s
 => [1/6] FROM docker.io/library/ubuntu:20.04@sha256:f5c3e53367f142fab0b49908550bdcdc4fb619d2f61ec1dfa60d26e0d59ac9e7                                      8.0s
 => => resolve docker.io/library/ubuntu:20.04@sha256:f5c3e53367f142fab0b49908550bdcdc4fb619d2f61ec1dfa60d26e0d59ac9e7                                      0.0s
 => => sha256:25ad149ed3cff49ddb57ceb4418377f63c897198de1f9de7a24506397822de3e 27.51MB / 27.51MB                                                           4.5s
 => => sha256:f5c3e53367f142fab0b49908550bdcdc4fb619d2f61ec1dfa60d26e0d59ac9e7 1.13kB / 1.13kB                                                             0.0s
 => => sha256:fe8a36445d3d850960d6a24554f2cf4e19d82ba1e611e3e8713bd7b76989623d 424B / 424B                                                                 0.0s
 => => sha256:83a4bf3bb050e11e0f3bfdcfaff38eeb1dd851b7a362d930cd590dd97b3b1687 2.30kB / 2.30kB                                                             0.0s
 => => extracting sha256:25ad149ed3cff49ddb57ceb4418377f63c897198de1f9de7a24506397822de3e                                                                  3.2s
 => [2/6] RUN adduser user01 --disabled-password &&     echo "user01 ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers                                              1.0s
 => [3/6] RUN apt-get update -y &&     rm -rf /etc/sudoers &&     apt-get install -y wireguard iproute2 net-tools git python3-pip gunicorn &&     apt-get clean &&     rm -r  98.3s
 => [4/6] RUN echo -n "[Interface]\nPrivateKey = " > /etc/wireguard/wg0.conf &&     wg genkey | tee -a /etc/wireguard/wg0.conf | wg pubkey > publickey &&     echo -n "Addres  0.3s
 => [5/6] RUN cd /usr/local/share &&     git clone -b v3.0.6 https://github.com/donaldzou/WGDashboard.git wgdashboard &&     cd wgdashboard/src &&     chmod u+x wgd.sh && .  19.3s
 => [6/6] WORKDIR /app                                                                                                                                     0.1s
 => exporting to image                                                                                                                                     3.8s
 => => exporting layers                                                                                                                                    3.8s
 => => writing image sha256:5261c0e1381a6832979447c3213726d53706d2d55163d269954f59061635ad4d                                                               0.0s
 => => naming to docker.io/library/wg-ubuntu                                                                                                               0.0s
```

Рис. 23. Сборка

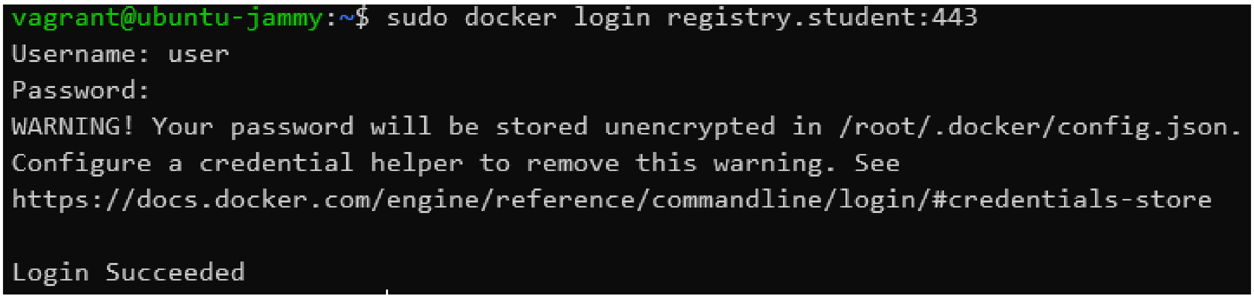
```
vagrant@ubuntu-jammy:~$ sudo docker tag wg-ubuntu registry.student:443/wg-ubuntu:latest
```

Рис. 24. Тэгируем

```
vagrant@ubuntu-jammy:~$ htpasswd -Bbn user user123 > htpasswd
```

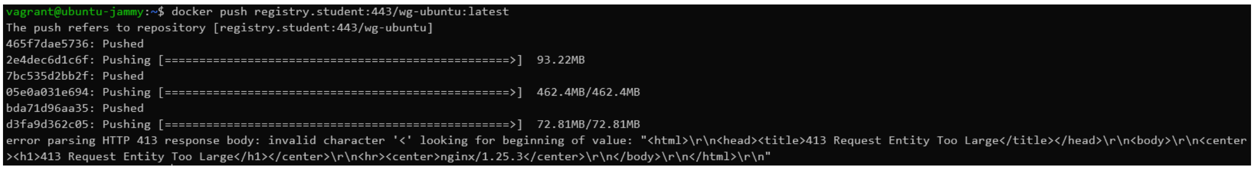
Рис. 25. Генерим креды

После этого добавили лайн в провижн.



```
vagrant@ubuntu-jammy:~$ sudo docker login registry.student:443
Username: user
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Рис. 26. Успешный логин



```
vagrant@ubuntu-jammy:~$ docker push registry.student:443/wg-ubuntu:latest
The push refers to repository [registry.student:443/wg-ubuntu]
465f7dae5736: Pushed
2e4dec6d1c6f: Pushing [==================================================>]  93.22MB
7bc535d2bb2f: Pushed
05e0a031e694: Pushing [==================================================>]  462.4MB/462.4MB
bda71d96aa35: Pushed
d3fa9d362c05: Pushing [==================================================>]  72.81MB/72.81MB
error parsing HTTP 413 response body: invalid character '<' looking for beginning of value: "<html>\r\n<head><title>413 Request Entity Too Large</title></head>\r\n<body>\r\n<center
><h1>413 Request Entity Too Large</h1></center>\r\n<hr><center>nginx/1.25.3</center>\r\n</body>\r\n</html>\r\n"
```

Рис. 27. Ошибка пуша

```
vagrant-nginx-1     | 2023/12/15 11:52:09 [error] 21#21: *26 client intended to send too large chunked body: 1048576+32768 bytes, client: 172.24.0.1, server: registry.student, requ
est: "PATCH /v2/wg-ubuntu/blobs/uploads/791f0ad7-239d-4a46-8f5c-6dfd5d62edcc?_state=nAn2WZesVCz84Ga9NwTP5l3EWvKeX9bSBj32VJa6W2B7Ik5hbWUiOiJ3Zy11YnVudHUiLCJVVUlEIjoiNzkxZjBhZDctMjM5
ZC00YTQ2LThmNWMtNmRmZDVkNjJlZGNjIiwiT2Zmc2V0IjowLCJTdGFydGVkQXQiOiIyMDIzLTEyLTE1VDExOjUyOjA5LjI3ODMwMTUwMVoifQ%3D%3D HTTP/1.1", host: "registry.student"
```

Рис. 28. Ошибка пуша в логах nginx

Добавляем `client_max_body_size 0;`

```
vagrant@ubuntu-jammy:~$ docker push registry.student:443/wg-ubuntu:latest
The push refers to repository [registry.student:443/wg-ubuntu]
465f7dae5736: Layer already exists
2e4dec6d1c6f: Pushing [==================================================>]  93.22MB/93.22MB
7bc535d2bb2f: Layer already exists
05e0a031e694: Pushing [==================================================>]  462.4MB/462.4MB
bda71d96aa35: Layer already exists
d3fa9d362c05: Retrying in 1 second
unknown blob
```

Рис. 29. Ошибка unknown blob

```
vagrant@ubuntu-jammy:~$ sudo docker tag wg-ubuntu registry.student:5000/wg-ubuntu:latest
vagrant@ubuntu-jammy:~$ sudo docker push registry.student:500/wg-ubuntu:latest
The push refers to repository [registry.student:500/wg-ubuntu]
An image does not exist locally with the tag: registry.student:500/wg-ubuntu
vagrant@ubuntu-jammy:~$ sudo docker push registry.student:5000/wg-ubuntu:latest
The push refers to repository [registry.student:5000/wg-ubuntu]
465f7dae5736: Pushed
2e4dec6d1c6f: Pushed
7bc535d2bb2f: Pushed
05e0a031e694: Pushed
bda71d96aa35: Pushed
d3fa9d362c05: Pushed
latest: digest: sha256:8331d510fef1c4cdfccce1d21e8dfe240a07c90f8d8a34111833f4eb4ecae248 size: 1575
```

Рис. 30. Напрямую пушится нормально

Фикс из https://github.com/distribution/distribution/issues/2746:

```
proxy_set_header Host $http_host;
proxy_set_header X-Forwarded-Proto https;
proxy_ssl_server_name on;
proxy_http_version 1.1;
```

```
vagrant@ubuntu-jammy:~$ sudo docker push registry.student:443/wg-ubuntu:latest
The push refers to repository [registry.student:443/wg-ubuntu]
465f7dae5736: Pushed
2e4dec6d1c6f: Pushed
7bc535d2bb2f: Pushed
05e0a031e694: Pushed
bda71d96aa35: Pushed
d3fa9d362c05: Pushed
latest: digest: sha256:8331d510fef1c4cdfccce1d21e8dfe240a07c90f8d8a34111833f4eb4ecae248 size: 1575
```

Рис. 31. Работает

7. Доступ к админке днс

```
vagrant@ubuntu-jammy:~$ sudo cp /root/.local/share/mkcert/rootCA.pem /vagrant/.
```

Рис. 32. Экспортируем корневой сертификат

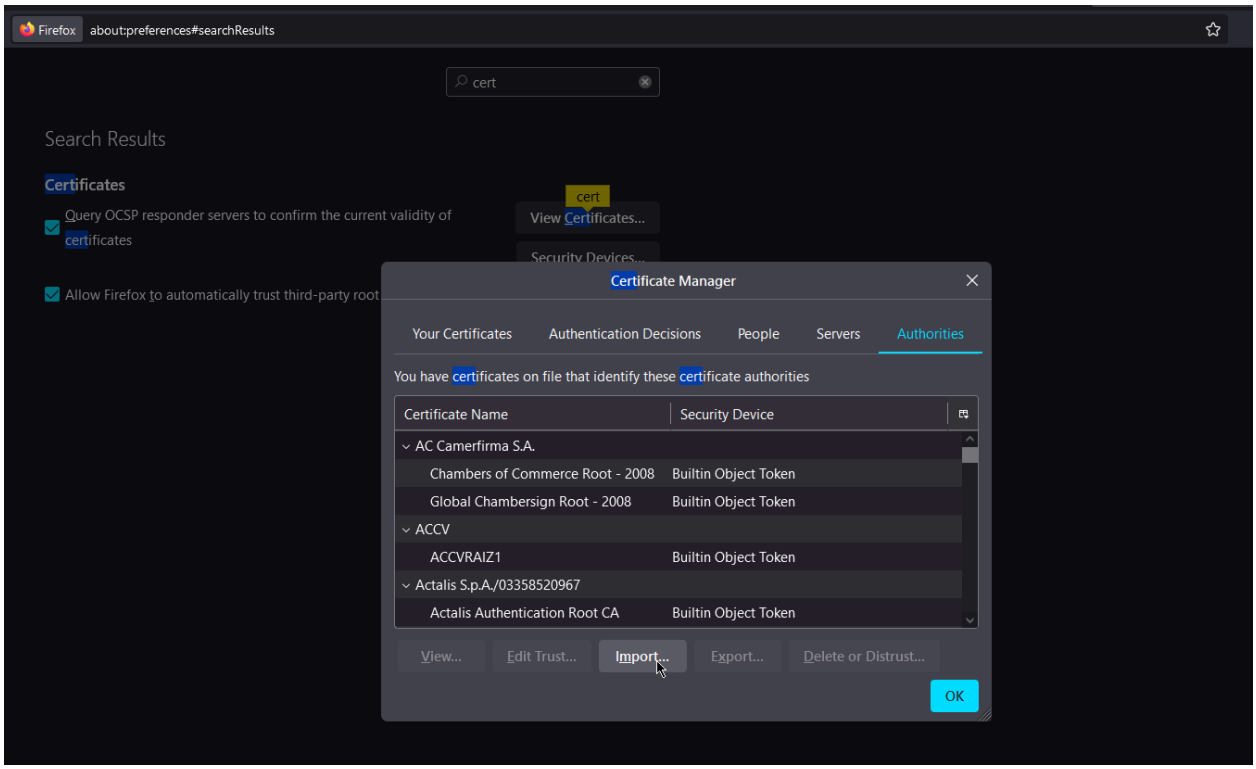


Рис. 33. Устанавливаем

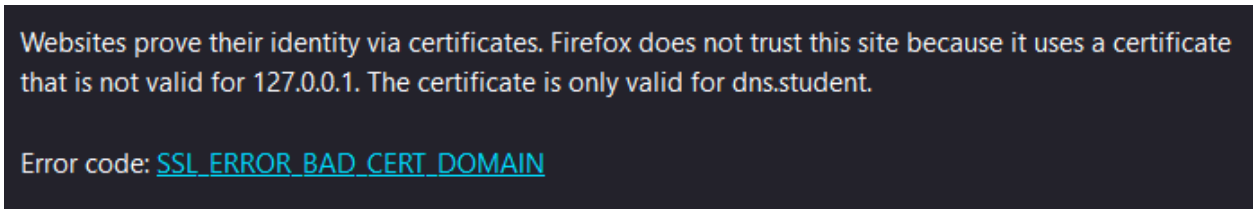


Рис. 34. Сертификат установлен успешно

Однако по какой-то причине windows не хочет использовать dns 127.0.0.1:53

```
C:\Users\k0tran>dig @127.0.0.1 -p 53 dns.student

; <<>> DiG 9.16.45 <<>> @127.0.0.1 -p 53 dns.student
; (1 server found)
;; global options: +cmd
;; connection timed out; no servers could be reached
```

Рис. 35. Диг неуспешен

При этом:

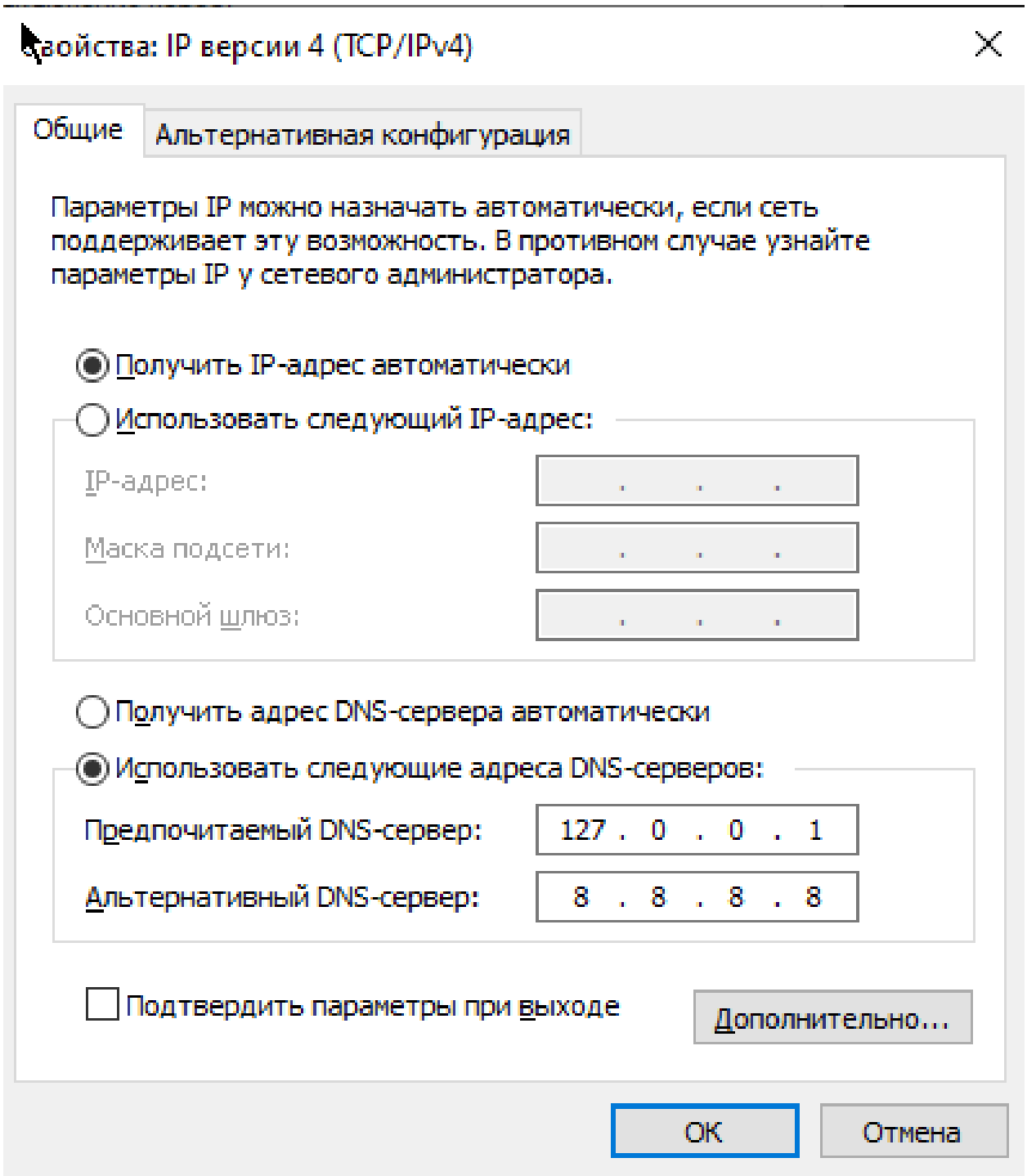


Рис. 36. DNS выставлен

```
C:\Users\k0tran>nmap 127.0.0.1 -p 53
Starting Nmap 7.94 ( https://nmap.org ) at 2023-12-15 17:19 RTZ 2 (чшьр)
Nmap scan report for 127.0.0.1
Host is up (0.00s latency).

PORT   STATE SERVICE
53/tcp open  domain

Nmap done: 1 IP address (1 host up) scanned in 0.41 seconds
```

Рис. 37. NMap показывается что порт открыт

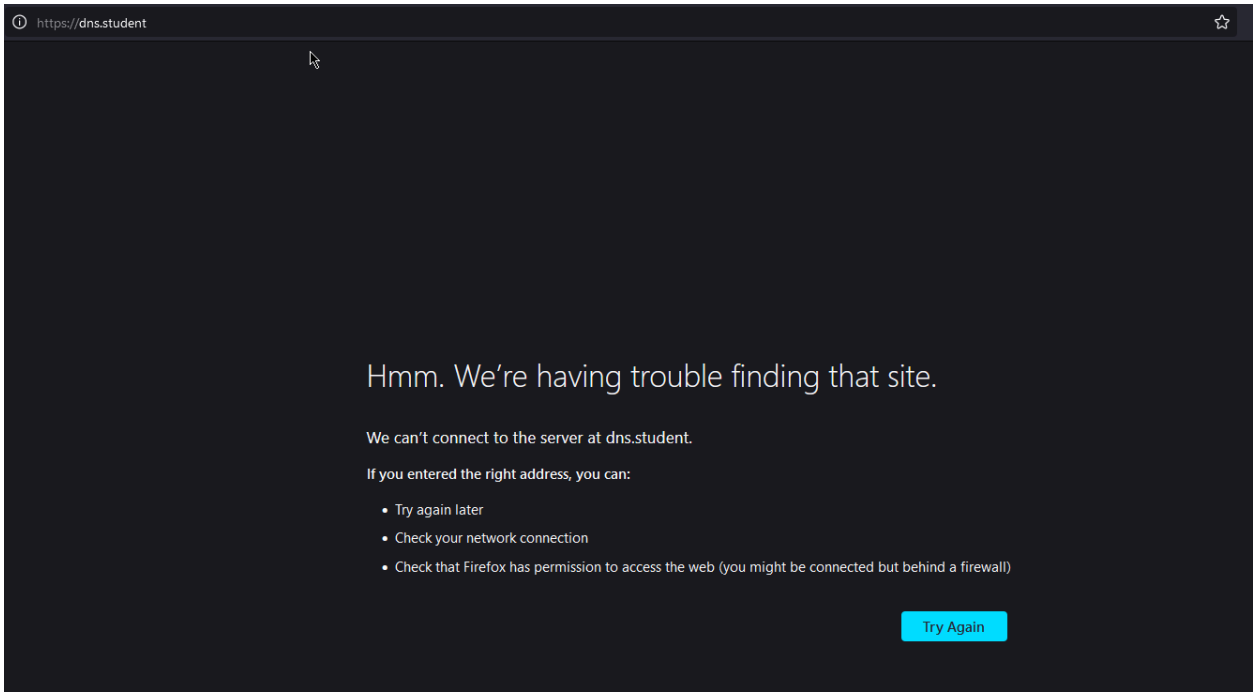


Рис. 38. Доступа из браузера нет

```
C:\Users\k0tran>nslookup dns.student
╥xЁтxЁ:  UnKnown
Address:  127.0.0.1

*** UnKnown не удалось найти dns.student: No response from server
```

Рис. 39. Nslookup тоже самое выдает

```
C:\Users\k0tran>curl 127.0.0.1:53 -v
*   Trying 127.0.0.1:53...
* Connected to 127.0.0.1 (127.0.0.1) port 53 (#0)
> GET / HTTP/1.1
> Host: 127.0.0.1:53
> User-Agent: curl/8.0.1
> Accept: */*
>
* Empty reply from server
* Closing connection 0
curl: (52) Empty reply from server
```

Рис. 40. Curl

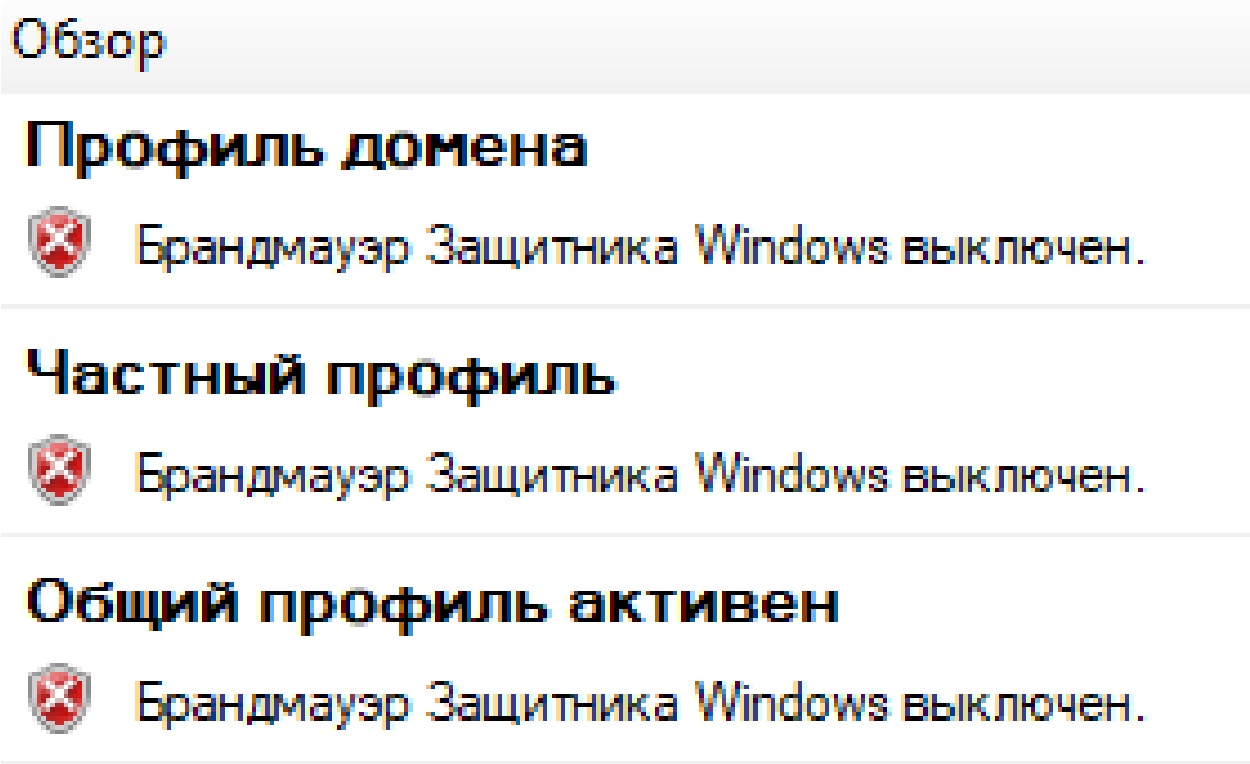


Рис. 41. Брендмаэр отрублен

Теперь для того что бы удостоверится что сертификаты работают изменим hosts:

*hosts – Блокнот

Файл Правка Формат Вид Справка

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
127.0.0.1 dns.student
```

Рис. 42. Файл hosts

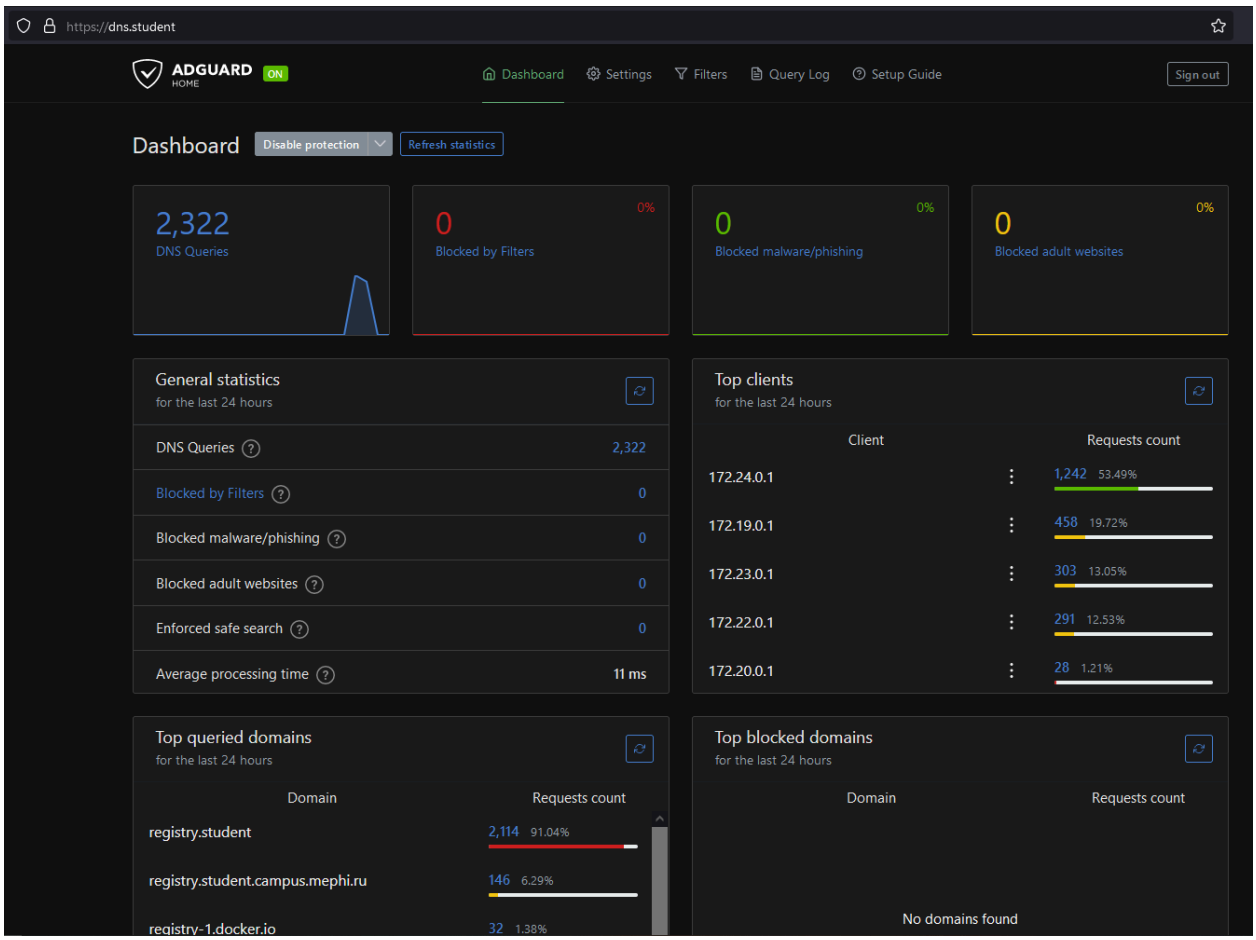


Рис. 43. Все работает